

Spring Data JPA系列：数据更新（Update）

上次通过《Spring Data JPA系列：使用@Modifying修改（Modifying queries）》介绍了数据更新的方式，这种更新方式会很很不方便，写的时候也比较麻烦，可以为更新密码、更新用户名等一些特殊的更新单独定义，但是对大多数数据操作是不方便的，比如我要更新一条有一百个字段的数据，这时候如果要通过Modifying方式就非常的不方便，因此，我们需要一种新的方式来解救。

通过阅读Spring-Data-JPA相关的文档和博客，找到了对应的解决方案，就是使用save()方法，经过测试，可用。

我们平时对save()方法的理解，大多是等同于insert()，主要是指新增一条数据，而JPA的save()方法包含了merge()的概念，就是说，如果save的对象不存在primary key或者primary key值在database内不存在的时候会新添加一条数据，如果primary key 存在并且primary key已经在database中存在，那就会依据primary key对该条数据进行更新，这是我们乐意见到的。

参考的文章：<https://stackoverflow.com/questions/11881479/how-do-i-update-an-entity-using-spring-data-jpa>

相关描述如下：

Identity of entities is defined by their primary keys. Since firstname and lastname are not parts of the primary key, you cannot tell JPA to treat Users with the same firstnames and lastnames as equal if they have different userIds.

So, if you want to update a User identified by its firstname and lastname, you need to find that User by a query, and then change appropriate fields of the object your found. These changes will be flushed to the database automatically at the end of transaction, so that you don't need to do anything to save these changes explicitly.

EDIT:

Perhaps I should elaborate on overall semantics of JPA. There are two main approaches to design of persistence APIs:

- insert/update approach. When you need to modify the database you should call methods of persistence API explicitly: you call insert to insert an object, or update to save new state of the object to the database.
- Unit of Work approach. In this case you have a set of objects managed by persistence library. All changes you make to these objects will be flushed to the database automatically at the end of Unit of Work (i.e. at the end of the current transaction in typical case). When you need to insert new record to the database, you make the corresponding object managed. Managed objects are identified by their primary keys, so that if you make an object with predefined primary key managed, it will be associated with the database record of the same id, and state of this object will be propagated to that record automatically.

JPA follows the later approach. save() in Spring Data JPA is backed by merge() in plain JPA, therefore it makes your entity managed as described above. It means that calling save() on an object with predefined id will update the corresponding database record rather than insert a new one, and also explains why save() is not called create().

标签：APP Spring Data

Spring Data

阅读	评论	喜欢
214	0	0
喜欢		



SpringForAll

文章	问答	粉丝
79	6	236
关注		